The Role of Retained-Layer Representations in Data Selection for Compressed Transformer Fine-Tuning

Victoria F. Mello[1], Flavio Soriano[1], Pedro B. Rigueira[1], Gisele L. Pappa[1], Wagner Meira Jr.[1], Washington Cunha[2], Marcos André Gonçalves[1]

[1] Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Brazil
{victoriaflores, flaviosoriano, pedrobacelar.rigueira, glpappa, meira, mgoncalv}@dcc.ufmg.br
[2] Instituto de Computação, Universidade Estadual de Campinas, Brazil
wcunha@unicamp.br

Abstract. Fine-tuning of Transformer-based models can be made more efficient by reducing the model, the training set, or both. However, layer pruning and training-instance selection are often treated as independent decisions, even though pruning layers changes the representation space in which examples are evaluated. This paper investigates whether data selection for fine-tuning compressed Transformer-based models should account for the representation space defined by the retained layers. We evaluate this hypothesis using BERT-base, selecting layers through kernel–target alignment (KTA) to favor task-relevant representations and centered kernel alignment (CKA) to penalize redundancy. The selected layers then define the representation kernel used for Nyström-based training-instance selection, followed by candidate-pool construction and balanced downselection. In experiments with BERT-base on SST-2 and QNLI across 12 seeds, the evaluated configuration achieves the highest average accuracy among the compressed variants considered in our experimental setting. On QNLI, it reaches $(61.36 \pm 6.44)\%$ accuracy and improves over BI+Nyström-Uniform by (6.25) paired percentage points, with 9/11 wins and BH-FDR ($\bar{0}.0205$). On SST-2, it reaches $(79.88 \pm 6.08)\%$ accuracy and remains competitive among the compressed configurations. Across both tasks, it reduces the aggregate FLOPs proxy by about 69% and peak CUDA memory by 13–19% relative to full BERT, although selection overhead still limits wall-clock gains. These results provide initial evidence that the representation space defined by the retained layers is a relevant consideration for data selection in compressed BERT fine-tuning.

CCS Concepts: • **Computing methodologies** → **Natural language processing**; *Neural networks*.

Keywords: centered kernel alignment, efficient fine-tuning, layer-aware data selection, Nyström approximation, pruning

1. INTRODUÇÃO

Os avanços obtidos por modelos *Transformer* em tarefas de processamento de linguagem natural (PLN) têm sido acompanhados pelo aumento de seu tamanho e de sua complexidade, tornando mais custosa sua adaptação a tarefas específicas [Men et al. 2025]. Mesmo a partir de modelos pré-treinados, o *fine-tuning* completo requer a execução de todas as camadas do encoder sobre um grande número de exemplos, com impactos sobre o tempo de processamento, o uso de memória e o consumo de recursos computacionais. Estratégias voltadas à redução desses custos são, portanto, relevantes para ampliar a acessibilidade e a sustentabilidade do uso de modelos neurais profundos.

Duas abordagens complementares consistem em reduzir o modelo [Xia et al. 2022] e o conjunto de dados empregado no *fine-tuning* [Cunha et al. 2023; Cunha et al. 2025]. Métodos de compressão atuam sobre a arquitetura por meio de poda, destilação ou adaptação de profundidade [Xu et al. 2020; Sanh et al. 2020; Xia et al. 2022], enquanto métodos de seleção de dados buscam preservar exemplos relevantes para otimização e generalização [Toneva et al. 2019; Paul et al. 2021] . Essas decisões, contudo, são frequentemente tratadas de forma independente, embora a representatividade

Agradecimentos: Este trabalho foi financiado e apoiado pelo INCT-TILDIAR (# 408490/2024-1), Google, NVIDIA, CIIA-Saúde, Ana Health, SEBRAE, EMBRAPII, FINEP, CAPES, CNPq, FAPEMIG e FAPESP.

de um exemplo dependa do espaço de características no qual ele é avaliado. Como a remoção de camadas altera esse espaço, investigamos se os exemplos de treinamento devem ser selecionados no espaço representacional definido pelas camadas que permanecem no modelo comprimido.

Para estudar essa questão, avaliamos uma configuração que seleciona camadas por uma combinação de *kernel–target alignment* (KTA) e *centered kernel alignment* (CKA) [Cortes et al. 2012; Kornblith et al. 2019]. As camadas preservadas definem o espaço de seleção dos exemplos por meio da média uniforme de seus kernels lineares, formando um kernel agregado. Esse kernel é usado para pontuar os exemplos com uma heurística inspirada em amostragem Nyström e *ridge leverage scores* [Williams and Seeger 2001; El Alaoui and Mahoney 2015; Musco and Musco 2017]. Avaliamos a configuração com BERT-base em SST-2 e QNLI, comparando-a com o modelo completo e com variantes comprimidas baseadas em seleção uniforme de exemplos, seleção aleatória de camadas e *Block Influence* (BI), uma medida proposta pelo ShortGPT para estimar a influência de cada bloco a partir da mudança entre sua entrada e sua saída [Men et al. 2025]. Os experimentos consideram 12 sementes, preservam 10 das 12 camadas do BERT e utilizam 50% dos dados de treinamento amostrados.

A configuração baseada em KTA+CKA+Nyström com balanceamento apresentou a maior acurácia média entre as variantes comprimidas avaliadas nas duas tarefas. O resultado mais expressivo foi observado em QNLI, com um ganho pareado médio de 6.25 pontos percentuais em relação à linha de base BI+Nyström-Uniform, além de uma redução de aproximadamente 69% no proxy de FLOPs e de 13–19% no pico de memória CUDA em relação ao BERT completo, embora a sobrecarga da etapa de seleção ainda limite ganhos de tempo fim a fim. Em conjunto, esses resultados fornecem evidências iniciais de que *a seleção de exemplos depende do espaço representacional definido pelas camadas preservadas do modelo comprimido.* Assim, as principais contribuições deste trabalho são: (i) investigar a hipótese de que a seleção de dados deve ser realizada no espaço representacional das camadas mantidas após a compressão; e (ii) apresentar uma avaliação inicial dessa hipótese considerando desempenho preditivo, custo computacional aproximado e uso de memória.

2. TRABALHOS RELACIONADOS

A redução do custo de modelos *Transformer* tem sido abordada por técnicas de destilação, poda, quantização e adaptação de profundidade [Cunha et al. 2025]. *BERT-of-Theseus* substitui progressivamente blocos do modelo professor [Xu et al. 2020]; *Movement Pruning* aprende padrões de esparsidade durante o *fine-tuning* [Sanh et al. 2020]; e *CoFi* realiza poda estruturada em múltiplas granularidades [Xia et al. 2022]. A estimativa da importância de componentes estruturais também tem sido explorada em outros contextos de recuperação de informação, como na ponderação de blocos de páginas Web. Nosso trabalho se aproxima mais da remoção estruturada de camadas após o pré-treinamento, particularmente de *ShortGPT*, que utiliza *Block Influence* (BI) para estimar a importância de cada camada [Men et al. 2025]. Diferentemente de BI, que define as camadas preservadas sem utilizar os rótulos da tarefa, avaliamos uma configuração que combina relevância supervisionada e diversidade antes da seleção de exemplos.

A seleção de dados busca reduzir o conjunto de treinamento preservando exemplos informativos para otimização e generalização. Entre as abordagens mais comuns estão critérios baseados em *forgetting events* [Toneva et al. 2019], *coresets* baseados em gradientes e medidas como EL2N e GraNd [Paul et al. 2021]. Representações semânticas também têm sido exploradas para organizar informação textual, inclusive em estruturas hierárquicas [Viegas et al. 2020]. Em aprendizado ativo para PLN, também há evidências de que a qualidade da seleção depende do estado representacional do modelo [Margatina et al. 2022], sugerindo que a importância de um exemplo é dependente do espaço em que ele é avaliado. Para conectar compressão de modelos e seleção de dados, utilizamos medidas de alinhamento entre kernels de representações. CKA mede similaridade entre representações neurais [Kornblith et al. 2019], enquanto KTA fornece uma medida supervisionada de alinhamento entre representações e rótulos [Cortes et al. 2012]. As camadas preservadas definem então o kernel usado para selecionar exemplos

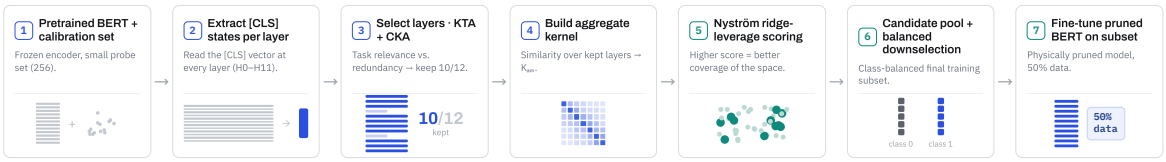

Fig. 1: Pipeline da configuração avaliada.

por uma heurística inspirada em métodos Nyström e *ridge leverage scores* [Williams and Seeger 2001; El Alaoui and Mahoney 2015; Musco and Musco 2017]. Assim, este trabalho conecta compressão de Transformers, seleção de dados e alinhamento por kernels ao investigar se a seleção de exemplos deve ser realizada no espaço representacional definido pelas camadas preservadas após a compressão.

3. MÉTODO

Esta seção descreve a estratégia de *fine-tuning* comprimido avaliada neste trabalho, na qual a seleção dos exemplos de treinamento depende do espaço representacional definido pelas camadas preservadas. A estratégia compreende quatro etapas: extração das representações de calibração, seleção de camadas por alinhamento de kernels, seleção de exemplos no espaço das camadas mantidas e *fine-tuning* do modelo fisicamente podado.

Considere um encoder Transformer pré-treinado com L blocos. Dado um conjunto de calibração

$$\mathcal{C} = \{(x_i, y_i)\}_{i=1}^{n_c},$$

extraímos as representações `[CLS]` após cada bloco ℓ, formando

$$H_\ell \in \mathbb{R}^{n_c \times d}, \qquad \ell \in \{0, \ldots, L-1\}.$$

O problema estudado consiste em escolher um subconjunto de camadas S, com $|S| = k_\ell$, e um subconjunto de exemplos de treinamento T, usado para adaptar o modelo comprimido. As camadas selecionadas são mantidas em sua ordem original, enquanto as demais são removidas antes do *fine-tuning*.

3.1 Seleção de camadas por alinhamento

Para cada matriz de representações H_ℓ, construímos um kernel linear

$$K(H_\ell) = H_\ell H_\ell^\top.$$

Esse kernel é centralizado antes do cálculo das medidas de alinhamento. O *kernel–target alignment* (KTA) compara o kernel de uma camada com um kernel construído a partir dos rótulos Y:

$$\mathrm{KTA}(\ell) = \mathrm{Align}(K_c(H_\ell), K_c(Y)).$$

Essa pontuação favorece camadas cujas representações estão mais alinhadas à tarefa supervisionada. Já o *centered kernel alignment* (CKA) compara os kernels de duas camadas:

$$\mathrm{CKA}(\ell, m) = \mathrm{Align}(K_c(H_\ell), K_c(H_m)),$$

sendo usado para medir redundância representacional.

A seleção de camadas é feita por uma heurística gulosa determinística. O conjunto S é inicializado com a última camada do encoder, de modo que a cabeça de classificação receba a saída de um bloco final de alto nível. Em cada passo seguinte, uma camada candidata $\ell \notin S$ recebe a pontuação

$$g(\ell \mid S) = \mathrm{KTA}(\ell) - \lambda \max_{m \in S} \mathrm{CKA}(\ell, m).$$

A primeira parcela favorece relevância para a tarefa; a segunda penaliza redundância com as camadas já selecionadas. A camada com maior pontuação é adicionada a S até que $|S| = k_\ell$.

3.2 Seleção de exemplos no espaço das camadas preservadas

Após selecionar as camadas, construímos um *kernel* agregado que representa o espaço no qual os exemplos de treinamento serão avaliados. Para o conjunto de camadas S, usamos a média uniforme dos kernels lineares:

$$K_{\text{agg}} = \frac{1}{|S|} \sum_{\ell \in S} H_\ell H_\ell^\top.$$

Essa escolha mantém a configuração simples e evita introduzir um peso adicional por camada. O ponto central é que o kernel usado para selecionar exemplos não é construído a partir de todas as camadas do modelo original, mas a partir do espaço representacional das camadas que permanecerão no modelo comprimido. Na implementação, o kernel agregado é centralizado e simetrizado; quando necessário, aplicamos uma pequena correção diagonal apenas para garantir estabilidade numérica.

A seleção de exemplos é feita por uma pontuação inspirada em amostragem Nyström e *ridge leverage scores*. Quando os candidatos coincidem com os exemplos de calibração, decompomos

$$\widetilde{K} = U \Sigma U^\top$$

e calculamos, para cada exemplo i,

$$\tau_i(\gamma) = \sum_j U_{ij}^2 \frac{\sigma_j}{\sigma_j + \gamma}.$$

Exemplos com maiores valores de τ_i são interpretados como mais relevantes para cobrir o espaço de kernel definido pelas camadas preservadas.

Nos experimentos reportados, a seleção é feita a partir de um conjunto de treinamento maior do que o conjunto de calibração. Seja C o conjunto de índices dos exemplos de calibração e k_{iC} o vetor de kernel entre um exemplo candidato i e os exemplos de calibração. A pontuação aproximada é calculada como

$$\widehat{\tau}_i = k_{iC}^c \left(\widetilde{K}_{CC} + \gamma I \right)^{-1} k_{Ci}^c.$$

As pontuações não negativas são normalizadas e usadas para formar um conjunto de candidatos. Em seguida, aplicamos uma etapa de seleção balanceada para compor o subconjunto final de treinamento, reduzindo o risco de selecionar exemplos de forma desbalanceada entre as classes.

3.3 Remoção física de camadas e orçamento de treinamento

Após a seleção, os blocos escolhidos passam a compor o encoder BERT, preservando a ordem que ocupavam no modelo pré-treinado. Os demais blocos são removidos da estrutura do modelo, resultando em um encoder com menor profundidade. Em seguida, o modelo comprimido é treinado no subconjunto de exemplos selecionados.

Para comparar configurações comprimidas de forma controlada, usamos um orçamento aproximado baseado em passos-camada:

$$B_{\text{proxy}} = b\, d\, L_{\text{active}}\, N_{\text{steps}},$$

em que b é o tamanho do lote, d é a dimensão oculta, L_{active} é o número de camadas ativas e N_{steps} é o número de passos de treinamento. Esse proxy não captura todos os custos reais do Transformer, como o termo quadrático da atenção, variação no comprimento das sequências, uso de memória e sobrecarga da seleção. Ainda assim, ele fornece uma forma simples de controlar a alocação experimental de treinamento e de analisar o compromisso entre acurácia, compressão e custo computacional aproximado.

4. CONFIGURAÇÃO EXPERIMENTAL

4.1 Tarefas, modelo e amostragem

A avaliação considera duas tarefas binárias do GLUE (*General Language Understanding Evaluation*), um conjunto de referência para a avaliação de modelos de compreensão de linguagem natural: SST-2 e QNLI [Socher et al. 2013; Wang et al. 2019]. SST-2 é uma tarefa de classificação de sentimento em nível de sentença, enquanto QNLI avalia a relação de inferência entre uma pergunta e uma sentença. Adotamos o `bert-base-uncased`, composto por 12 camadas e dimensão oculta 768, como arquitetura-base por ser um modelo amplamente utilizado em avaliações no GLUE e por oferecer uma configuração padronizada para comparar estratégias de poda e seleção de dados. [Devlin et al. 2019].

Os experimentos principais usam 12 sementes fixas. Para cada semente, os subconjuntos de treinamento e validação são amostrados deterministicamente a partir das divisões originais do GLUE, para garantir reprodutibilidade e comparação direta. As comparações pareadas usam os mesmos subconjuntos dentro de cada semente, o que permite comparar as configurações sob as mesmas condições. A configuração comprimida principal e as linhas de base comprimidas preservam 10 das 12 camadas do BERT e usam 50% do conjunto de treinamento amostrado.

A seleção de camadas e exemplos usa um conjunto de calibração com 256 exemplos. A configuração principal solicita um conjunto candidato com multiplicador 4.0 antes da seleção balanceada final; nas execuções reportadas, seu tamanho é limitado pelo próprio conjunto de calibração. Para o *fine-tuning*, adotamos *batch size* 16, comprimento máximo de 128 tokens e taxa de aprendizado (2×10^{-5}), valores compatíveis com configurações amplamente utilizadas em experimentos com BERT e GLUE [Devlin et al. 2019; Mosbach et al. 2021; Di Liello et al. 2022]. Os parâmetros específicos do método foram mantidos fixos em todas as execuções: ($\lambda = 0.5$), que controla o compromisso entre relevância supervisionada e redundância na seleção de camadas, e ($\gamma = 10^{-3}$), que regulariza a estimativa dos *ridge leverage scores* [El Alaoui and Mahoney 2015; Musco and Musco 2017]. O treinamento é realizado com o `Trainer` da biblioteca Hugging Face, usando um número fixo de passos, sem *early stopping* ou seleção do melhor checkpoint. As execuções foram realizadas em GPU com precisão fp32.

4.2 Configurações comparadas

Comparamos a configuração principal baseada em KTA+CKA+Nyström com quatro variantes. A primeira é o modelo completo, que mantém todas as 12 camadas do BERT e usa todo o conjunto de treinamento amostrado. Essa configuração fornece o teto de acurácia esperado no cenário experimental. A segunda referência é uma variante sequencial simples, que seleciona camada por KTA+CKA e depois aplica seleção uniforme por Nyström no espaço das camadas preservadas, sem construção de conjunto candidato e sem seleção balanceada. Essa variante isola a ideia central de usar o espaço das camadas mantidas, mas remove as etapas adicionais da configuração principal. A terceira referência usa camadas selecionadas aleatoriamente, preservando também a camada final, seguida pela mesma família de seleção por Nyström. Essa comparação verifica se qualquer subconjunto de camadas seria suficiente para orientar a seleção de exemplos ou se a escolha do espaço representacional é de fato relevante. A quarta referência usa Block Influence (BI) para selecionar 10 camadas, preservando a camada final por construção, e depois aplica seleção uniforme por Nyström. Essa linha de base permite comparar a configuração proposta com uma estratégia estruturada de poda de camadas que não usa os rótulos da tarefa na definição do espaço representacional.

4.3 Métricas e análise estatística

A métrica principal é a acurácia, seguindo o protocolo de avaliação do GLUE para SST-2 e QNLI e considerando que ambas as tarefas apresentam distribuição aproximadamente balanceada entre as classes [Wang et al. 2019]. Reportamos média e desvio padrão sobre as 12 sementes. Para as com-

Table I: Resultados principais em SST-2 e QNLI (12 sementes). Acurácia reportada como média ± desvio padrão. BI possui uma execução inválida em cada tarefa; por isso, sua média considera 11 execuções.

Dataset	Configuração	Camadas	Dados	Válidas	Acurácia (%)
SST-2	Full BERT	12	100%	12/12	85.81 ± 2.56
	KTA+CKA+Nyström-Bal.	10	50%	12/12	79.88 ± 6.08
	KTA+CKA+Nyström	10	50%	12/12	79.62 ± 5.16
	BI+Nyström-Uniform	10	50%	11/12	78.91 ± 6.73
	RandomLayers+Nyström	10	50%	12/12	76.53 ± 4.39
QNLI	Full BERT	12	100%	12/12	69.24 ± 5.36
	KTA+CKA+Nyström-Bal.	10	50%	12/12	61.36 ± 6.44
	KTA+CKA+Nyström	10	50%	12/12	58.72 ± 4.91
	RandomLayers+Nyström	10	50%	12/12	58.17 ± 4.31
	BI+Nyström-Uniform	10	50%	11/12	56.11 ± 3.48

parações entre métodos comprimidos, usamos diferenças pareadas por semente, aproveitando o fato de que cada configuração é avaliada sobre os mesmos subconjuntos de dados dentro de uma mesma semente. Quando uma execução de uma linha de base é marcada como colapsada, essa semente é removida apenas das comparações pareadas envolvendo a respectiva linha de base; isso ocorre uma vez para BI+Nyström-Uniform em SST-2 e uma vez em QNLI. Além da acurácia, reportamos medidas associadas à eficiência computacional, incluindo tempo de treinamento, tempo de seleção, tempo total de parede, energia, pico de memória CUDA e um proxy de FLOPs. Todas as execuções ocorreram na mesma máquina e em ambiente isolado, reduzindo interferências nas medições. Essas medidas são usadas para analisar o compromisso entre desempenho e custo computacional. Em particular, o proxy de FLOPs e o pico de memória ajudam a caracterizar a redução obtida pela compressão, enquanto o tempo total de parede evidencia o custo adicional introduzido pela etapa de seleção.

5. RESULTADOS

5.1 Resultados principais em SST-2 e QNLI

A Tabela I resume os resultados principais em SST-2 e QNLI considerando 12 sementes. Como esperado, o BERT completo mantém a maior acurácia média nas duas tarefas, pois usa todas as 12 camadas e 100% dos dados de treinamento amostrados. O objetivo deste trabalho, entretanto, não é superar o modelo completo, mas avaliar se uma configuração comprimida que seleciona exemplos no espaço das camadas mantidas pode permanecer competitiva ao reduzir camadas e dados. Entre as configurações comprimidas, KTA+CKA+Nyström-Bal. apresenta a melhor média nas duas tarefas. O resultado mais claro aparece em QNLI: essa configuração supera BI+Nyström-Uniform por +6.25 pontos percentuais pareados, RandomLayers+Nyström por +3.19 pontos e a variante KTA+CKA+Nyström por +2.64 pontos. Em SST-2, as diferenças são menores, mas a configuração principal ainda apresenta a melhor média entre os métodos comprimidos e fica +3.35 pontos acima de RandomLayers+Nyström.

5.2 Comparações pareadas

A Tabela II apresenta comparações pareadas entre a configuração principal e as demais configurações comprimidas. As diferenças são calculadas por semente, usando os mesmos subconjuntos de dados dentro de cada par. Valores positivos indicam vantagem para KTA+CKA+Nyström-Bal.

A comparação mais forte é observada em QNLI contra BI+Nyström-Uniform: KTA+CKA+Nyström-Bal. vence em 9 dos 11 pares válidos, com diferença média de +6.25 pontos percentuais e BH-FDR = 0.0205. As demais comparações em QNLI também são positivas em média e em número de vitórias, embora com evidência estatística mais fraca após correção. Em SST-2, os resultados devem ser interpretados com mais cautela: a configuração principal permanece competitiva, mas as diferenças pareadas não caracterizam uma vitória estatística isolada.

Table II: Comparações pareada contra KTA+CKA+Nyström-Bal. Diferenças positivas favorecem a configuração principal. Comparações com BI usam 11 pares porque uma execução de BI falhou em cada tarefa.

Dataset	Comparador	Pares	Vitórias/derrotas	Diferença média (pp)	BH-FDR
SST-2	KTA+CKA+Nyström	12	6/6	+0.26	0.4631
SST-2	RandomLayers+Nyström	12	8/4	+3.35	0.1384
SST-2	BI+Nyström-Uniform	11	5/6	+2.41	0.2329
QNLI	KTA+CKA+Nyström	12	9/3	+2.64	0.0881
QNLI	RandomLayers+Nyström	12	8/4	+3.19	0.0663
QNLI	BI+Nyström-Uniform	11	9/2	+6.25	0.0205

Table III: Resumo de eficiência para Full BERT e KTA+CKA+Nyström-Bal. O tempo total inclui a etapa de seleção quando aplicável. FLOPs são proxies agregados sobre seleção e *fine-tuning*.

Dataset	Configuração	Acurácia	Total (s)	Treino (s)	Seleção (s)	Energia (kWh)	CUDA MB	FLOPs proxy
SST-2	Full BERT	0.8581	14.59	13.53	0.00	0.001294	2127.67	3.317×10^9
SST-2	KTA+CKA+Nyström-Bal.	0.7988	20.88	11.25	8.66	0.001416	1848.01	1.006×10^9
QNLI	Full BERT	0.6924	23.36	21.87	0.00	0.002285	2536.82	4.541×10^{10}
QNLI	KTA+CKA+Nyström-Bal.	0.6136	26.18	19.13	5.35	0.002405	2062.51	1.411×10^{10}

5.3 Compromisso entre acurácia e eficiência

Além da acurácia, avaliamos medidas associadas à eficiência computacional: tempo total de parede, tempo de treinamento, tempo de seleção, energia, pico de memória CUDA e um proxy agregado de FLOPs. A Tabela III compara o BERT completo com a configuração KTA+CKA+Nyström-Bal. nas duas tarefas principais. A configuração comprimida reduz o proxy total de FLOPs em aproximadamente 69% nas duas tarefas em relação ao BERT completo. Também reduz o pico de memória CUDA em 13.1% em SST-2 e 18.7% em QNLI. Embora o tempo de *fine-tuning* seja menor a etapa de seleção, executada em CPU, introduz uma sobrecarga anterior ao treinamento. Como resultado, o tempo total de parede aumenta em 43.1% em SST-2 e 12.1% em QNLI. Essa sobrecarga, contudo, não corresponde a maior ocupação da GPU, recurso computacional mais oneroso e proibitivo. Assim, os resultados caracterizam um compromisso entre acurácia, redução do custo computacional aproximado, menor uso de memória e menor tempo de treinamento em GPU, sem representar aceleração fim a fim.

6. DISCUSSÕES, LIMITAÇÕES E TRABALHOS FUTUROS

Os resultados fornecem evidências iniciais de que o espaço representacional usado para selecionar exemplos influencia a qualidade do subconjunto empregado no *fine-tuning* comprimido de *Transformers*. Comparada às variantes baseadas em camadas aleatórias e em Block Influence, a configuração KTA+CKA+Nyström apresentou resultados mais consistentes, especialmente em QNLI, sugerindo que combinar relevância supervisionada e diversidade representacional produz um espaço mais adequado para a seleção de exemplos. Em SST-2, os ganhos foram menores, indicando que a evidência ainda deve ser interpretada como inicial, e não como demonstração de superioridade geral.

O objetivo deste trabalho não é estabelecer KTA+CKA+Nyström como a melhor combinação possível de poda de camadas e seleção de dados, mas investigar se essas decisões devem ser tratadas de forma integrada. O escopo experimental é restrito a duas tarefas, um único modelo, uma fração fixa de dados e uma única configuração de profundidade. Além disso, a comparação principal combina a escolha das camadas preservadas e do espaço usado para selecionar exemplos; um experimento fatorial 2×2 permitiria isolar melhor esses efeitos. Do ponto de vista prático, a configuração avaliada reduz substancialmente o proxy de FLOPs e o pico de memória CUDA em relação ao BERT completo, embora a sobrecarga da etapa de seleção ainda impeça interpretar os resultados como aceleração fim a fim. Avaliações futuras devem considerar um conjunto mais amplo de tarefas, arquiteturas e protocolos de medição de custo, incluindo tempo de seleção, treinamento, inferência e consumo energético.

7. CONCLUSÃO

Este artigo investigou se a seleção de exemplos para o *fine-tuning* de *Transformers* comprimidos deve considerar o espaço representacional definido pelas camadas preservadas após a poda. Para isso, avaliamos uma configuração que combina seleção de camadas por KTA+CKA com seleção de exemplos por Nyström sobre o kernel induzido pelas camadas mantidas. Os resultados em SST-2 e QNLI fornecem evidências iniciais de que essa escolha metodológica influencia a qualidade da seleção de dados. A configuração KTA+CKA+Nyström com balanceamento apresentou a maior acurácia média entre as variantes comprimidas avaliadas, além de reduzir o proxy de FLOPs em cerca de 69% e o pico de memória CUDA em 13–19% em relação ao BERT completo, embora a sobrecarga da seleção ainda limite ganhos de tempo fim a fim. Como trabalhos futuros, pretendemos ampliar a avaliação para outras tarefas, arquiteturas e métodos de compressão e seleção de dados. Em conjunto, os resultados sugerem que o espaço representacional definido pelas camadas preservadas merece consideração explícita na seleção de dados para *fine-tuning* eficiente de *Transformers*.

REFERENCES

Cortes, C., Mohri, M., and Rostamizadeh, A. Algorithms for learning kernels based on centered alignment. *Journal of Machine Learning Research* 13 (28): 795–828, 2012.

Cunha, W., França, C., Fonseca, G., Rocha, L., and Gonçalves, M. A. An effective, efficient, and scalable confidence-based instance selection framework for transformer-based text classification. In *SIGIR*. pp. 665–674, 2023.

Cunha, W., Moreo Fernández, A., Esuli, A., Sebastiani, F., Rocha, L., and Gonçalves, M. A. A noise-oriented and redundancy-aware instance selection framework. *ACM TOIS* 43 (2): 1–33, 2025.

Cunha, W., Rocha, L., and Gonçalves, M. A. Inteligência artificial sustentável baseado em engenharia de dados, aprendizado de máquina e transferência de conhecimento para processamento de linguagem natural. In *SBBD*, 2025.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the NAACL*. pp. 4171–4186, 2019.

Di Liello, L., Gabburo, M., and Moschitti, A. Effective pretraining objectives for transformer-based autoencoders. In *Findings of EMNLP 2022*. Association for Computational Linguistics, pp. 5533–5547, 2022.

El Alaoui, A. and Mahoney, M. W. Fast randomized kernel ridge regression with statistical guarantees. In *Advances in Neural Information Processing Systems*. Vol. 28. pp. 775–783, 2015.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. Similarity of neural network representations revisited. In *ICML*. Vol. 97. PMLR, pp. 3519–3529, 2019.

Margatina, K., Barrault, L., and Aletras, N. On the importance of effectively adapting pretrained language models for active learning. In *ACL*. pp. 825–836, 2022.

Men, X., Xu, M., Zhang, Q., Wang, B., Lin, H., Lu, Y., Han, X., and Chen, W. ShortGPT: Layers in large language models are more redundant than you expect. In *Findings of ACL 2025*, 2025.

Mosbach, M., Andriushchenko, M., and Klakow, D. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines, 2021.

Musco, C. and Musco, C. Recursive sampling for the Nyström method. In *NeurIPS*. Vol. 30, 2017.

Paul, M., Ganguli, S., and Dziugaite, G. K. Deep learning on a data diet: Finding important examples early in training. In *Advances in Neural Information Processing Systems*. Vol. 34. pp. 20596–20607, 2021.

Sanh, V., Wolf, T., and Rush, A. M. Movement pruning: Adaptive sparsity by fine-tuning. In *Advances in Neural Information Processing Systems*. Vol. 33. pp. 20378–20389, 2020.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP 2013*. pp. 1631–1642, 2013.

Toneva, M., Sordoni, A., Tachet des Combes, R., Trischler, A., Bengio, Y., and Gordon, G. J. An empirical study of example forgetting during deep neural network learning. In *ICLR*, 2019.

Viegas, F., Cunha, W., Gomes, C., Pereira, A., Rocha, L., and Goncalves, M. CluHTM - semantic hierarchical topic modeling based on CluWords. In *Proceedings of the 58th ACL*. pp. 8138–8150, 2020.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*, 2019.

Williams, C. K. I. and Seeger, M. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems*. Vol. 13. MIT Press, pp. 682–688, 2001.

Xia, M., Zhong, Z., and Chen, D. Structured pruning learns compact and accurate models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. pp. 1513–1528, 2022.

Xu, C., Zhou, W., Ge, T., Wei, F., and Zhou, M. BERT-of-Theseus: Compressing BERT by progressive module replacing. In *Proceedings of the 2020 EMNLP*. Association for Computational Linguistics, pp. 7859–7869, 2020.